

# Case Statements

# What You Will Learn

---

- Case statements

# Case Statements

---

- Alternative to if statements
  - `if [ "$VAR" = "one" ]`
  - `elif [ "$VAR" = "two" ]`
  - `elif [ "$VAR" = "three" ]`
  - `elif [ "$VAR" = "four" ]`
- May be easier to read than complex if statements.

```
case "$VAR" in
    pattern_1)
        # Commands go here.
        ;;
    pattern_N)
        # Commands go here.
        ;;
esac
```

```
case "$1" in
    start)
        /usr/sbin/sshd
        ;;
    stop)
        kill $(cat /var/run/sshd.pid)
        ;;
esac
```

```
case "$1" in
    start)
        /usr/sbin/sshd
        ;;
    stop)
        kill $(cat /var/run/sshd.pid)
        ;;
    *)
        echo "Usage: $0 start|stop" ; exit 1
        ;;
esac
```

```
case "$1" in
    start|START)
        /usr/sbin/sshd
        ;;
    stop|STOP)
        kill $(cat /var/run/sshd.pid)
        ;;
    *)
        echo "Usage: $0 start|stop" ; exit 1
        ;;
esac
```

```
read -p "Enter y or n: " ANSWER
case "$ANSWER" in
    [yY] | [yY][eE][sS])
        echo "You answered yes."
        ;;
    [nN] | [nN][oO])
        echo "You answered no."
        ;;
    *)
        echo "Invalid answer."
        ;;
esac
```



```
read -p "Enter y or n: " ANSWER
case "$ANSWER" in
    [yY]*)
        echo "You answered yes."
        ;;
    *)
        echo "You answered something else."
        ;;
esac
```

# Summary

---

- Can be used in place of **if** statements.
- Patterns can include wildcards.
- Multiple pattern matching using a pipe.